Libraries Used

1.Numerical and Data Handling Libraries:

- NumPy: Fundamental package for numerical computations in Python.

```
import numpy as np
from numpy import *
```

- Pandas: Provides data structures and data analysis tools.

```
import pandas as pd
```

2. Operating System Utilities:

- os: Miscellaneous operating system interfaces.

```
import os
```

3. **File Handling**:

- zipfile: Work with ZIP archives.

```
from zipfile import *
import zipfile
```

4. Image Processing:

- OpenCV: Library of programming functions mainly aimed at real-time computer vision.

```
import cv2
```

- PIL: Python Imaging Library, adds image processing capabilities.

```
from PIL import Image
```

5. Data Visualization:

- Matplotlib: Plotting library for creating static, animated, and interactive visualizations.

```
import matplotlib.pyplot as plt
%matplotlib inline
```

- Seaborn: Statistical data visualization based on Matplotlib.

```
import seaborn as sns
```

6. Progress Bar:

- tqdm: A fast, extensible progress bar for loops and file processing.

```
from tqdm import tqdm
```

7. Machine Learning and Deep Learning:

- TensorFlow and Keras: Open-source platform for machine learning, used for implementing neural networks.

```
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers import Dense, Flatten, MaxPooling2D, Convolution2D
from tensorflow.keras.applications import InceptionV3
from tensorflow.keras.layers import Dense, GlobalAveragePooling2D, Dropout
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau
from tensorflow.keras.models import load_model
from keras.preprocessing.image import ImageDataGenerator, load_img,  array_to_img, img_to_array
```

8.Scikit-Learn:

- train_test_split: Function for splitting data arrays into two subsets: training data and testing data.

```
from sklearn.model_selection import train_test_split
```

- confusion_matrix: Compute confusion matrix to evaluate the accuracy of a classification.

```
from sklearn.metrics import confusion_matrix
```

9.Google Colab Specific:

- Google Drive Mounting: For accessing files on Google Drive.

```
from google.colab import drive
```

drive.mount('/content/drive')

- Colab Patches: Tools for displaying images in Google Colab.

from google.colab.patches import cv2_imshow

10. IPython Display:

- IPython Display: Provides tools for displaying rich media (like images) in IPython environments.

python

from IPython.display import display

Summary

This comprehensive list covers all the libraries used for:

- Numerical and data handling (`NumPy`, `Pandas`).
- File system operations (`os`, `zipfile`).
- Image processing and handling (`cv2`, `PIL`).
- Data visualization (`Matplotlib`, `Seaborn`).
- Progress tracking (`tqdm`).
- Machine learning and neural networks (`TensorFlow`, `Keras`, `Scikit-Learn`).
- Google Colab specific functionalities (`google.colab`, `IPython`).

The main libraries in Django, flask

Python =3.1.0

pillow==10.3.0

Django==5.0.6

requests==2.32.3

pip file

Flask==3.0.3

Flask-Cors==4.0.1

tensorflow-intel==2.16.1

keras==3.3.3

numpy==1.26.4

joblib==1.4.2

opencv-python==4.10.0.82

virtualenv==20.26.2

uvicorn==0.30.1

Reference

1] Y. Fan, K. Zhao, Z.-L. Shi, et al., Bat coronaviruses in China, Viruses 11 (2019) 210–223.

[2] S. Chauhan, Comprehensive review of coronavirus disease 2019 (COVID-19), Biomed. J. 43 (2020) 334–340.

[3] B. Abhijit, et al., A deep learning based approach for automatic detection of COVID-19 cases using chest X-ray images, Biomed. Signal Process Control 71 (Part B) (2022) 103182

[4] D. Dong, Z. Tang, S. Wang, et al., The role of imaging in the detection and management of COVID-19: a review, IEEE Rev. Biomed. Eng. 14 (2020) 16–29.

[5] V.M. Corman, O. Landt, M. Kaiser, et al., Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR, Euro Surveill. 25 (2020) 23–30.

[6] T. Ai, Z. Yang, H. Hou, et al., Correlation of chest CT and RT-PCR testing in coronavirus disease 2019 (COVID-19) in China: a report of 1014 cases, Radiology 296 (2020) E32–E40.

[7] T. Iwasawa, M. Sato, T. Yamaya, et al., Ultra-high-resolution computed tomography can demonstrate alveolar collapse in novel coronavirus (COVID-19) pneumonia, Jpn.

J. Radiol. 38 (2020) 394–398.
[8] L.A. Rousan, E. Elobeid, M. Karrar, et al., Chest x-ray findings and temporal lung changes in patients with COVID-19 pneumonia, BMC Pulm. Med. 20 (2020) 1–9.
[9] O. Attallah, X. Ma, Bayesian neural network approach for determining the risk of re-intervention after endovascular aortic aneurysm repair, Proc. IME H J. Eng. Med. 228 (2014) 857–866.
[10] A. Karthikesalingam, O. Attallah, X. Ma, et al., An artificial neural network stratifies the risks of Reintervention and mortality after endovascular aneurysm repair; a retrospective observational study, PLoS One 10 (2015), e0129024.
[11] O. Attallah, MB-AI-His: histopathological diagnosis of pediatric medulloblastoma and its subtypes via AI, Diagnostics 11 (2021) 359–384
[12] M.A. Ozdemir, G.D. Ozdemir, O. Guren, Classification of COVID-19 electrocardiograms by using hexaxial feature mapping and deep learning, BMC Med. Inf. Decis. Making 21 (2021) 1–20.
[13] C. Shorten, T.M. Khoshgoftaar, A survey on image data augmentation for deep learning, J. Big Data 6 (2019) 1–48
[14] S.U. Amin, M. Alsulaiman, G. Muhammad, et al., Deep Learning for EEG motor imagery classification based on multi-layer CNNs feature fusion, Future Generat. Comput. Syst. 101 (2019) 542–554.
[15] Q. Xu, Z. Wang, F. Wang, et al., Multi-feature fusion CNNs for Drosophila embryo of interest detection, Phys. Stat. Mech. Appl. 531 (2019), 121808
[16] Q. Zhang, H. Li, Z. Sun, et al., Deep feature fusion for Iris and periocular biometrics on mobile devices, IEEE Trans. Inf. Forensics Secur. 13 (2018) 2897–2912.
[17] L. Hongtao, Z. Qinchuan, Applications of deep convolutional neural network in computer vision, J. Data Acquis. Process. 31 (2016) 1–17.